2.4 Platform Dependencies

Due to the close coupling of the A.G.P. and main memory subsystem, there are several behaviors of the A.G.P. that will likely end up being platform dependent. While the objective of any specification is to minimize such differences, it is apparent that several are probable among A.G.P. corelogic and platform implementations. This should not, however, have as much impact as it would in other buses for two reasons:

- The A.G.P. is a point-to-point[3] connection, intended for use by a 3D graphics accelerator only, and,
- Due to performance issues (Section 2.3), an A.G.P. graphics master will likely need to be optimized to a specific subset of platform or corelogic implementations anyway.

The purpose of this section is to identify by example some of the areas where behavioral differences are likely, and accordingly establish the scope of this interface specification.

As one example of potential variation, consider the two corelogic architectures shown in Figure 2-3. An integrated approach, typical of desktop and volume systems, is shown on the left, and a symmetric multiprocessor partitioning, typical of MP servers, is shown on the right.

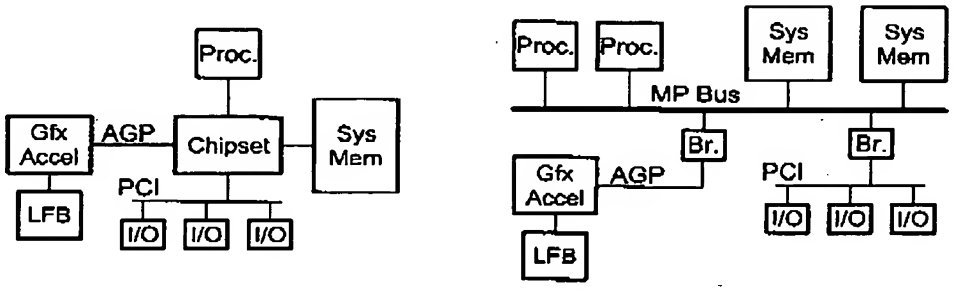

Figure 2-3: Different Corelogic Architectures

The following items are examples of areas where behavioral differences between these or other implementations could well occur:

GART Implementation

For various reasons, different systems may opt for different GART table implementations and layouts. This, however, is not visible since the actual table implementation is abstracted to a common API by the HAL or miniport driver supplied with the corelogic.

Coherency with Processor Cache

Due to the high potential access rate on the A.G.P., it is not advisable from a performance perspective to snoop all accesses. Selective snooping in an integrated corelogic architecture presents serious queue

[3] This means that active communication can only occur between two A.G.P. agents that reside on the interface, where one agent is referred to as the A.G.P. target and the other the A.G.P. master. The simplest implementation is to only have two devices attached to the bus. Attaching more than two devices to the interface is not precluded as long as there is only one active master and one active target. Any other device must not respond to or interfere with the interface operation. When more than two devices are attached to the interface, the system designer is responsible to ensure that all requirements of this interface specification are met, since the component and/or add-in card designer has no control how the devices are used.

A. Terminology

A.G.P. bus	Abbreviation for PCI/A.G.P. bus.
A.G.P. master	An A.G.P. compliant master interface that is capable of generating A.G.P. pipelined read/write operations per this interface specification.
A.G.P. operation	Memory read/write operation subject to A.G.P. ordering rules and protocol. A.G.P. operations that are initiated by **PIPE#** or **SBA** bus.
A.G.P. port	Connection point on a chipset where an A.G.P. compliant master may be attached.
API	Application Programming Interface.
Bus 0	Compatibility PCI bus *(where the ISA bridge resides).*
Bus *n*	PCI/A.G.P. bus. $n = 1$ when no PCI to PCI Bridges present on Bus 0.
Chipset	Motherboard chipset that provides connections to: Host Bus, Compatibility PCI bus, and A.G.P. interface.
DirectX	Microsoft API's for accessing graphics and audio hardware.
DirectDraw	Microsoft graphics API.
Display surface	Memory area containing graphics data object.
Fence	Means of synchronizing A.G.P. write operations with subsequent A.G.P. read operations.
Flush	Operation that makes an A.G.P. compliant target's accesses to system graphics memory visible to other parts of the system.
A.G.P. target	An A.G.P. compliant target interface that is capable of interpreting A.G.P. addresses (e.g., the chipset) per this interface specification.
Local graphics memory	Memory local to the graphics controller.
Non-prefetchable memory	PCI registers that have side effects.
Prefetchable Memory	PCI memory and memory mapped registers that are free from side-effects.
Uncacheable Memory	Host caching protocol used on I/O operations, non-prefetchable regions or prefetchable regions not supported by hardware coherency.
Ordering rules	Rules specifying when the effect of a read or write operation can be observed by another operation.
Paging	Movement of data between disk and other memory levels of the virtual memory system.
Pipelining	A.G.P. read/write operations use a *split transaction* like paradigm where one or more addresses may be transferred during one bus operation and data is transferred during another.
POST or POST Code	Initialization software executed out of the startup ROM.
SPI	System Programming Interface.